

The Cost of Parameterized Reachability in Mobile Ad Hoc Networks

Giorgio Delzanno¹, Arnaud Sangnier²,
Riccardo Traverso¹, and Gianluigi Zavattaro³

¹ University of Genova, Italy

² LIAFA, Univ Paris Diderot, Paris Cité Sorbonne, CNRS, France

³ University of Bologna, INRIA - FOCUS Research Team, Italy

Abstract. We investigate the impact of spontaneous movement in the complexity of verification problems for an automata-based protocol model of networks with selective broadcast communication. We first consider reachability of an error state and show that parameterized verification is decidable with polynomial complexity. We then move to richer queries and show how the complexity changes when considering properties with negation or cardinality constraints.

1 Introduction

Selective broadcast communication is often used in networks in which individual nodes have no precise information about the underlying connection topology (e.g. ad hoc wireless networks). As shown in [13,10,11,16,17,4], this type of communication can naturally be specified in models in which a network configuration is represented as a graph and in which individual nodes run an instance of a given protocol specification. A protocol typically specifies a sequence of control states in which a node can send a message (emitter role), wait for a message (receiver role), or perform an update of its internal state. Selective broadcast communication is modeled as a simultaneous update of the state of the emitter node and of the states of its neighbors.

Already at this level of abstraction, verification of protocols with selective broadcast communication turns out to be a very difficult task. A formal account of this problem is given in [3,4], where the *control state reachability problem* is proved to be undecidable in an automata-based protocol model in which configurations are arbitrary graphs. The control state reachability problem consists in verifying the existence of an initial network configuration (with unknown size and topology) that may evolve into a configuration in which at least one node is in a given control state. If such a control state represents a protocol error, then this problem naturally expresses (the complement of) a safety verification task in a setting in which nodes have no information a priori about the size and connection topology of the underlying network.

In presence of spontaneous movement, i.e., non-deterministic reconfigurations of the network during an execution, control state reachability becomes decidable

[3]. In this paper we focus on the complexity of different types of parameterized reachability problems in presence of spontaneous movement. More precisely, we consider reachability queries defined over assertions that: (PRP) check the presence or absence of control states in a given configuration generated by an initial configuration of arbitrary size, and (CRP) cardinality queries that check the exact number of occurrences of control states in a reachable configuration (the counterpart of classical reachability). The first and the second problem require, at least in principle, the exploration of an infinite-state space. Indeed they are formulated for arbitrary initial configurations. The latter is inherently finite-state. Despite of it, we first show that reachability queries for constraints that only check for the presence of a control state can be checked in polynomial time. When considering both constraints for checking presence and absence of control states the problem turns out to be NP-complete. Finally, we show that the problem becomes PSPACE-complete for cardinality queries.

Related Work. Perfect synchronous semantics for broadcast communication have been proposed in [14,16,17,5]. Semantics that take into consideration interferences and conflicts during a transmission have been proposed in [8,10,11,12]. To our knowledge, parameterized verification has not been studied in previous work on formal models of ad hoc networks [14,16,17,13,5,7,8,10,11,12]. Finally, decidability issues for broadcast communication in unstructured concurrent systems have been studied, e.g., in [6], whereas verification of unreliable communicating FIFO systems have been studied, e.g., in [1].

2 A Model for Mobile Ad Hoc Networks

2.1 Syntax and semantics

Our model for mobile ad hoc networks is defined in two steps. We first define graphs used to denote network configurations and then define protocols running on each node. The label of a node denotes its current control state. Finally, we give a transition system for describing the interaction of a vicinity during the execution of the same protocol on each node.

Definition 1. A Q -graph is a labeled undirected graph $\gamma = \langle V, E, L \rangle$, where V is a finite set of nodes, $E \subseteq V \times V$ is a finite set of edges, and L is a labeling function from V to a set of labels Q .

We use $L(\gamma)$ to represent all the labels present in γ (i.e. the image of the function L). The nodes belonging to an edge are called the *endpoints* of the edge. For an edge $\langle u, v \rangle$ in E , we use the notation $u \sim_\gamma v$ and say that the vertices u and v are adjacent one to another in the graph γ . We omit γ , and simply write $u \sim v$, when it is made clear by the context.

Definition 2. A process is a tuple $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$, where Q is a finite set of control states, Σ is a finite alphabet, $R \subseteq Q \times (\{!!a, ??a \mid a \in \Sigma\}) \times Q$ is the transition relation, and $Q_0 \subseteq Q$ is a set of initial control states.

The label $!!a$ [resp. $??a$] represents the capability of broadcasting [resp. receiving] a message $a \in \Sigma$. For $q \in Q$ and $a \in \Sigma$, we define the set $R_a(q) = \{q' \in Q \mid \langle q, ??a, q' \rangle \in R\}$ which contains the states that can be reached from the state q when receiving the message a . We assume that $R_a(q)$ is non empty for every a and q , i.e. nodes always react to broadcast messages. Local transitions (denoted by the special label τ) can be derived by using a special message m_τ such that $\langle q, ??m_\tau, q' \rangle$ implies $q' = q$ for every $q \in Q$ (i.e. receivers do not modify their local states). In the following, if for some state $q \in Q$ and message $a \in \Sigma$ we omit the definition of transitions of the form $\langle q, ??a, q' \rangle$, we implicitly assume the existence of only one such transition that does not change the state (i.e. $q' = q$).

Given a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$, a configuration of the corresponding Mobile Ad Hoc Network (MAHN) is a Q -graph and an initial configuration is a Q_0 -graph. We use \mathcal{C} [resp. \mathcal{C}_0] to denote the set of configurations [resp. initial configurations] associated to \mathcal{P} . Note that even if Q is finite, there are infinitely many possible configurations (the number of Q -graphs). We assume that each node of the graph is a process that runs a common predefined protocol defined by a communicating automaton with a finite set Q of control states. Communication is achieved via selective broadcast, which means that a broadcasted message is received by the nodes which are adjacent to the sender. We next formalize the above intuition.

Given a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$, a MAHN is defined by the transition system $MAHN(\mathcal{P}) = \langle \mathcal{C}, \rightarrow, \mathcal{C}_0 \rangle$ where the transition relation $\rightarrow \subseteq \mathcal{C} \times \mathcal{C}$ is such that: for $\gamma, \gamma' \in \mathcal{C}$ with $\gamma = \langle V, E, L \rangle$, we have $\gamma \rightarrow \gamma'$ iff $\gamma' = \langle V, E', L' \rangle$ and one of the following conditions holds:

- Broadcast** $E' = E$ and $\exists v \in V$ s.t. $\langle L(v), !!a, L'(v) \rangle \in R$ and $L'(u) \in R_a(L(u))$ for every $u \sim v$, and $L(w) = L'(w)$ for any other node w .
- Movement** $E' \subseteq V \times V$ and $L = L'$.

We use \rightarrow^* to denote the reflexive and transitive closure of \rightarrow .

2.2 Parameterized Reachability Problems

A natural class of verification problems for MAHN consists in determining whether there exists an initial configuration from which a configuration respecting some constraints can be reached. In this work, the constraints are boolean combination of atoms which allow to state the presence or the absence of a control state in a configuration. Given a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$, a *constraint* over \mathcal{P} is defined by the following grammar: $\varphi ::= \#q \geq 1 \mid \#q = 0 \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$ with $q \in Q$. We denote by CC the class of constraints and by $CC[\geq 1]$ the class of constraints in which atomic propositions have only the form $\#q \geq 1$ (there exists at least one occurrence of q). Given a configuration γ the satisfaction relation \models for constraints is defined by (we omit boolean cases defined as usual): $\gamma \models \#q \geq 1$ iff $q \in L(\gamma)$ and $\gamma \models \#q = 0$ iff $q \notin L(\gamma)$.

The parameterized reachability problem (PRP) can then be stated as follows:

Input: A process \mathcal{P} with $MAHN(\mathcal{P}) = \langle \mathcal{C}, \rightarrow, \mathcal{C}_0 \rangle$ and a constraint φ .

Output: Yes, if $\exists \gamma_0 \in \mathcal{C}_0$ and $\gamma_1 \in \mathcal{C}$ s.t. $\gamma_0 \rightarrow^* \gamma_1$ and $\gamma_1 \models \varphi$.

If the answer to this problem is yes, we will write $\mathcal{P} \models \Diamond \varphi$. We use the term *parameterized* to remark that the initial configuration is not fixed a priori. In fact, the only constraint that we put on the initial configuration is that the nodes have labels taken from Q_0 without any information on their number or connection links. As a special case we can define the control state reachability problems studied in [3] as the PRP for the simple constraint $\#q \geq 1$ (i.e. is there an initial configuration that can reach a configuration in which the state q is exposed?).

We also remark that according to the semantics, the number of nodes stays constant in each execution starting from the same initial configuration. As a consequence, when fixing the initial configuration γ_0 , we obtain finitely many possible reachable configurations. Thus, checking if there exists γ_1 reachable from a given γ_0 s.t. $\gamma_1 \models \varphi$ for a constraint φ is a decidable problem.

On the other hand, checking the parameterized version of the reachability problem is in general more difficult. Indeed, in [3], it is proved that PRP for simple constraints of the form $\#q \geq 1$ is undecidable when deleting the movement rule from the semantics (i.e. nodes communicate via selective broadcast but the connectivity graph never changes during a computation). In [3], it is also proved that PRP for the same class of simple constraints is decidable. However, the proposed decidability proof is based on a reduction to the problem of coverability in Petri nets which is known to be EXPSpace-complete [18,19]. Since no lower-bound was provided, the precise complexity of PRP with simple constraints was left as an open problem that we close in this paper by showing that it is PTIME-complete.

3 PRP restricted to constraints in $CC[\geq 1]$

In this section, we study PRP restricted to $CC[\geq 1]$. Note that this class of constraints allow to characterize configurations in which a given set of control states is present but they cannot express neither the absence of states nor the number of their occurrences. We first give a lower bound for this problem.

Proposition 1. *PRP restricted to $CC[\geq 1]$ is PTIME-hard.*

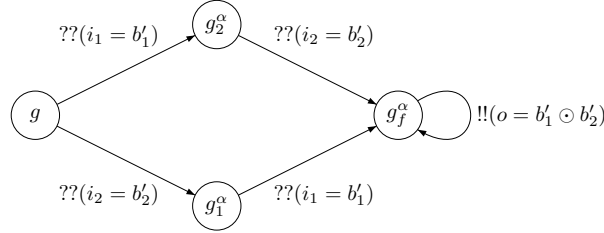
Proof. The proof is based on a LOGSPACE-reduction from the Circuit Value Problem (CVP) which is known to be PTIME-complete [9]. CVP is defined as follows: given an acyclic Boolean circuit with k input variables, m boolean gates (of type and, or, not), a single output variable and a truth assignment for the input variables, is the value of the output equal to a given boolean value?

Assume an instance of CVP C with input/output/intermediate value names taken from a finite set VN . We denote by $v_1, \dots, v_k \in VN$ the inputs and by $v \in VN$ the output. Furthermore, each gate g is represented by its signature $g(\odot, i_1, i_2, o)$ with $i_1, i_2, o \in VN$ and $\odot \in \{\vee, \wedge\}$ or by $g(\neg, i, o)$ with $i, o \in VN$. Finally, let $b_1, \dots, b_k \in \{true, false\}$ be a truth assignment for the inputs and $b \in \{true, false\}$ the value for the output to be tested.

The process \mathcal{P}_C associated to C has two types of initial states: q_0 (init nodes), and g (gate nodes) for each gate g of C . A node in state q_0 broadcasts (an arbitrary number of) messages that model the initial assignments to input variables. Since the assignment is fixed, broadcasting these messages several times (or receiving them from different initial nodes) does not harm the correctness of the encoding. When receiving an evaluation for their inputs (from an initial node or another gate node), a gate node evaluates the corresponding boolean function and then repeatedly broadcasts the value of the corresponding output. Since C is acyclic, once computed, the output value remains always the same (i.e. recomputing it does not harm). Finally, reception of a value v for output z sends a q_0 node into state ok . Reachability of an output value v reduces then to PRP for the process \mathcal{P}_C with ok the control state to be reached.

Formally, the process rules are defined as follows. For $i \in \{1, \dots, k\}$, we have rules $\langle q_0, !!(v_i = b_i), q_0 \rangle$ and $\langle q_0, ??(v = b), ok \rangle$. They model the assignment of value v_i to input x_i and reception of output value v .

For gate $g(\odot, i_1, i_2, o)$ and for each assignment $\alpha = \langle b'_1, b'_2 \rangle$ (with $b'_1, b'_2 \in \{true, false\}$) of values to $\langle i_1, i_2 \rangle$ (a constant number for each gate), we associate the following subprotocol:



(Self-loops associated to receptions for which there are no explicit rules are omitted). We use a similar encoding for a *not* gate.

Consider now the resulting process $\mathcal{P}_C = \langle Q, \Sigma, R, \{q_0\} \cup \{g \mid g \text{ is a gate in } C\} \rangle$ with corresponding transition system $MAHN = \langle \mathcal{C}, \rightarrow, \mathcal{C}_0 \rangle$. We have that there exists $\gamma \in \mathcal{C}_0$ and $\gamma' \in \mathcal{C}$ s.t. $\gamma \rightarrow^* \gamma'$ and $\gamma' \models \#ok \geq 1$ iff b is the value for v in C with input values b_1, \dots, b_k . \square

We now show that PRP restricted to $CC[\geq 1]$ is in PTIME. The main idea to obtain this result lies in the fact that we can compute in polynomial time the set of control states that appear in the reachable configurations. The construction is based on the following key points. We first observe that, in order to decide if control state q can be reached, we can focus our attention on initial complete graphs (i.e. graphs in which all pairs of nodes are connected). Indeed, spontaneous movement can be applied to non-deterministically transform a topology into any other one. Another key observation is that if a configuration γ can be reached from an initial configuration γ_0 , then for any natural k , there exists a complete graph which is reachable from an initial configuration γ'_0 and in which each of the control states appearing in γ appears at least k times. The initial configuration γ'_0 is obtained by replicating k -times the initial graph γ_0 . The

replicated parts are then connected in all possible ways (to obtain a connected graph). We can then use spontaneous movement to activate and deactivate the different subparts in order to mimick k parallel executions of the original system. For what concerns constraints in $\text{CC}[\geq 1]$ this property of PRP avoids the need of counting the occurrences of states. We just have to remember which states can be generated by repeatedly applying process rules. By exploiting the above

Algorithm 1 Computing the set of control states reachable in a MAHN

Input : $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$ a process

Output : $S \subseteq Q$ the set of reachable control states in $\text{MAHN}(\mathcal{P})$

```

 $S := Q_0$ 
 $oldS := \emptyset$ 
while  $S \neq oldS$  do
   $oldS := S$ 
  for all  $\langle q_1, !!a, q_2 \rangle \in R$  such that  $q_1 \in oldS$  do
     $S := S \cup \{q_2\} \cup \{q' \in Q \mid \langle q, ??a, q' \rangle \in R \wedge q \in oldS\}$ 
  end for
end while

```

mentioned observations, when defining the decision procedure for checking control state reachability we can take the following assumptions: (i) forget about the topology underlying the initial configuration; (ii) forget about the number of occurrences of control states in a configuration (if it is reached once, it can be reached an arbitrary number of times by considering larger initial configurations as explained before); (iii) consider a single symbolic path in which at each step we apply all possible rules whose preconditions can be satisfied in the current set and then collect the resulting set of computed states.

We now formalize the previous observations. Let $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$ be a process with $\text{MAHN}(\mathcal{P}) = \langle \mathcal{C}, \rightarrow, \mathcal{C}_0 \rangle$ and let $\text{Reach}(\mathcal{P})$ be the set of reachable control states equals to $\{q \in Q \mid \exists \gamma \in \mathcal{C}_0. \exists \gamma' \in \mathcal{C}. \text{ s.t. } \gamma \rightarrow^* \gamma' \text{ and } q \in L(\gamma')\}$. We will now prove that Algorithm 1 computes $\text{Reach}(\mathcal{P})$. Let S be the result of the Algorithm 1 (note that this algorithm necessarily terminates because the **while**-loop is performed at most $|Q|$ times). We have then the following lemma.

Lemma 1. *The two following properties hold:*

- (i) *There exist two configurations $\gamma_0 \in \mathcal{C}_0$ and $\gamma \in \mathcal{C}$ such that $\gamma_0 \rightarrow^* \gamma$ and $L(\gamma) = S$.*
- (ii) *$S = \text{Reach}(\mathcal{P})$.*

Proof. We first prove (i). We denote by S_0, S_1, \dots, S_n the content of S after each iteration of the loop of the Algorithm 1. We recall that a graph $\gamma = \langle V, E, L \rangle$ is complete if $\langle v, v' \rangle \in E$ or $\langle v', v \rangle \in E$ for all $v, v' \in V$. We will now consider the following statement: for all $j \in \{0, n\}$, for all $k \in \mathbb{N}$, there exists a complete graph $\gamma_{j,k} = \langle V, E, L \rangle$ in \mathcal{C} verifying the two following points:

1. $L(\gamma_{j,k}) = S_j$ and for each $q \in S_j$, the set $\{v \in V \mid L(v) = q\}$ has more than k elements (i.e. for each element q of S_j there are more than k nodes in $\gamma_{j,k}$ labeled with q),
2. there exists $\gamma_0 \in \mathcal{C}_0$ such that $\gamma_0 \rightarrow^* \gamma_{j,k}$.

To prove this statement we reason by induction on j . First, for $j = 0$, the property is true, because for each $k \in \mathbb{N}$, the graph $\gamma_{0,k}$ corresponds to the complete graphs where each of the initial control states appears at least k times. We now assume that the property is true for all naturals smaller than j (with $j < n$) and we will show it is true for $j + 1$. We consider E the set $\{\langle \langle q_1, !!a, q_2 \rangle, \langle q, ??a, q' \rangle \rangle \in R \times R \mid q_1, q \in S_j\}$ and M its cardinality. Let $k \in \mathbb{N}$ and let $N = k + 2 * k * M$. We consider the graph $\gamma_{j,N}$ where each control state present in S_j appears at least N times (such a graph exists by the induction hypothesis). From $\gamma_{j,N}$, we build the graph $\gamma_{j+1,k}$ obtained by repeating k times the following operations:

- for each pair $\langle \langle q_1, !!a, q_2 \rangle, \langle q, ??a, q' \rangle \rangle \in E$, select a node labeled by q_1 and one labeled by q and update their label respectively to q_2 and q' (this simulates a broadcast from the node labeled by q_1 received by the node labeled q in the configuration in which all the other nodes have been disconnected thanks to the movement and reconnected after). Note that the two selected nodes can communicate because the graph is complete.

By applying these rules it is then clear that $\gamma_{j,N} \rightarrow^* \gamma_{j+1,k}$ and also that $\gamma_{j+1,k}$ verifies the property 1 of the statement. Since by induction hypothesis, we have that there exists $\gamma_0 \in \mathcal{C}_0$ such that $\gamma_0 \rightarrow^* \gamma_{j,N}$, we also deduce that $\gamma_0 \rightarrow^* \gamma_{j+1,k}$, hence the property 2 of the statement also holds. From this we deduce that (i) is true.

To prove (ii), from (i) we have that $S \subseteq \text{Reach}(\mathcal{P})$ and we now prove that $\text{Reach}(\mathcal{P}) \subseteq S$. Let $q \in \text{Reach}(\mathcal{P})$. We show that $q \in S$ by induction on the minimal length of an execution path $\gamma_0 \rightarrow^* \gamma$ such that $\gamma_0 \in \mathcal{C}_0$ and $q \in L(\gamma)$. If the length is 0 then $q \in Q_0$ hence also $q \in S$. Otherwise, let $\gamma' \rightarrow \gamma$ be the last transition of the execution. We have that there exists $q_1 \in L(\gamma')$ such that $\langle q_1, !!a, q \rangle \in R$ [or $q_1, q_2 \in L(\gamma')$ such that $\langle q_1, !!a, q_3 \rangle, \langle q_2, ??a, q \rangle \in R$]. By induction hypothesis we have that $q_1 \in S$ [or $q_1, q_2 \in S$]. By construction, we can conclude that also $q \in S$. \square

Since constraints in $\text{CC}[\geq 1]$ check only the presence of states and do not contain negation, given a configuration γ and a constraint φ in $\text{CC}[\geq 1]$ such that $\gamma \models \varphi$, we also have that $\gamma' \models \varphi$ for every γ' such that $L(\gamma) \subseteq L(\gamma')$. Moreover, given a process \mathcal{P} , by definition of $\text{Reach}(\mathcal{P})$ we have that $L(\gamma) \subseteq \text{Reach}(\mathcal{P})$ for every reachable configuration γ , and by Lemma 1 there exists a reachable configuration γ_f such that $L(\gamma_f) = \text{Reach}(\mathcal{P})$. Hence, to check $\mathcal{P} \models \Diamond \varphi$ it is sufficient to verify whether $\gamma_f \models \varphi$ for such a configuration γ_f . This can be done algorithmically as follows: once the set $\text{Reach}(\mathcal{P})$ is computed, check if the boolean formula obtained from φ by replacing each atomic constraint of the form $\#q \geq 1$ by *true* if $q \in \text{Reach}(\mathcal{P})$ and by *false* otherwise is valid. This allows us to state the following theorem.

Theorem 1. *PRP restricted to $CC[\geq 1]$ is PTIME-complete.*

Proof. The lower bound is given by Proposition 1. To obtain the upper bound, it suffices to remark that the Algorithm 1 is in PTIME since it requires at most $|Q|$ iterations each one requiring at most $|R|^2$ look-ups (of active broadcast/receive transitions) for computing new states to be included, and also that evaluating the validity of a boolean formula can be done in polynomial time. \square

4 Complexity for PRP

In this section we study the decidability and complexity of PRP for constraints in CC. The main difference with the problem studied in the previous section lies in the fact that now the constraints have the ability to specify that a given control state is not present in a configuration (using atomic constraints of the form $\#q = 0$). Authorizing this kind of atomic constraints leads to a complexity jump as stated by the following proposition whose proof can be found in Appendix.

Proposition 2. *PRP for constraints in CC is NP-hard.*

Proof. The proof is based on a reduction of the boolean satisfiability problem (SAT) which is known to be NP-complete. Let Φ be a boolean formula in conjunctive normal form over the set of variables $V = \{v_1, \dots, v_k\}$. We define a process \mathcal{P} with initial state q_0 and the following set of rules $R = \{\langle q_0, \tau, v \rangle \mid v \in V\} \cup \{\langle q_0, \tau, \bar{v} \rangle \mid v \in V\}$. From Φ , we build a constraint $\varphi \wedge \psi$ where φ is the formula obtained from Φ by replacing each positive literal v by $\#v \geq 1$ and each negative literal $\neg v$ by $\#\bar{v} \geq 1$ and $\psi = \bigwedge_{i=1}^k (\#v_i \geq 1 \wedge \#\bar{v}_i = 0) \vee (\#v_i = 0 \wedge \#\bar{v}_i \geq 1)$. The former constraint is the natural encoding of the input propositional formula whereas the latter assigns a consistent interpretation to the control state labels v_i and \bar{v}_i as assignments to the propositional variable v_i . The constraint $\varphi \wedge \psi$ is a formula in CC.

A node in the initial state q_0 makes a guess for the boolean valuation of a variable v by moving to state v [resp. to \bar{v}] if the associated chosen value is *true* [resp. *false*]. The formula ψ ensures that no contradictory valuation is generated by stating that for each variable v in V only one type of control state v or \bar{v} is chosen. Assume that the formula Φ is satisfiable and let $\{b_1, \dots, b_k\} \in \{\text{true}, \text{false}\}^k$ be an interpretation over the variables $\{v_1, \dots, v_k\}$ that satisfies it. From an initial configuration γ_0 with k nodes, it is possible to reach a configuration γ such that $\gamma \models \psi$ and for all $1 \leq i \leq k$ if $b_i = \text{true}$ then $\gamma' \models \#v_i \geq 1$ else $\gamma' \models \#v_i = 0$. $\gamma \models \varphi \wedge \psi$ clearly holds here. Vice versa, if there exists a computation that reaches a configuration that satisfies $\varphi \wedge \psi$, then we have $m \geq k$ nodes whose labels correspond to a consistent interpretation of the variables in V and which satisfies Φ . \square

We will now give an algorithm in NP to solve PRP for constraints in CC. As for Algorithm 1, this new algorithm works on sets of control states. The algorithm works in two main phases. In a first phase it generates an increasing sequence of

sets of control states that can be reached in the considered process definition. At each step the algorithm adds the control states obtained from the application of the process rules to the current set of labels. Unlike the Algorithm 1, this new algorithm does not merge different branches, i.e. application of distinct rules may lead to different sequences of sets of control states. In a second phase the algorithm only removes control states applying again process rules in order to reach a set of control states that satisfies the given constraint.

Algorithm 2 Solving PRP for constraints in CC

Input : $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$ a process and φ a constraint over \mathcal{P} in CC

Output : Does $\mathcal{P} \models \Diamond\varphi$?

 Guess $S_0, \dots, S_m, T_1, \dots, T_n \subseteq Q$ with $m, n \leq |Q|$

 If $S_0 \not\subseteq Q_0$ return NO

for all $i \in \{0, \dots, m-1\}$ **do**

 If $S_{i+1} \notin \text{postAdd}(\mathcal{P}, S_i)$ return NO

end for

$T_0 = S_m$

for all $i \in \{0, \dots, n-1\}$ **do**

 If $T_{i+1} \notin \text{postDel}(\mathcal{P}, T_i)$ return NO

end for

 If T_n satisfies φ return YES else return NO

For a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$ and a set $S \subseteq Q$, we define the operator $\text{postAdd}(\mathcal{P}, S) \subseteq 2^Q$ as follows: $S' \in \text{postAdd}(\mathcal{P}, S)$ if and only if the two following conditions are satisfied: (i) $S \subseteq S'$ and (ii) for all $q' \in S' \setminus S$, there exists a rule $\langle q, !!a, q' \rangle \in R$ such that $q \in S$ (q' is produced by a broadcast) or there exist rules $\langle p, !!a, p' \rangle$ and $\langle q, ??a, q' \rangle \in R$ such that $q, p \in S$ and $p' \in S'$ (q' is produced by a reception). In other words, all the states in $S' \in \text{postAdd}(\mathcal{P}, S)$ are either in S or states obtained from the application of broadcast/reception rules to labels in S . Similarly, we define the operator $\text{postDel}(\mathcal{P}, S) \subseteq 2^Q$ as follows: $S' \in \text{postDel}(\mathcal{P}, S)$ if and only if $S' \subseteq S$ and one of the following conditions hold: either $S \setminus S' = \emptyset$ or $[S \setminus S' = \{q\}]$ and there exists a rule $\langle q, !!a, q' \rangle \in R$ such that $q' \in S'$ or $[S \setminus S' = \{q\}]$ and there exist two rules $\langle p, !!a, p' \rangle, \langle q, ??a, q' \rangle \in R$ such that $p, p', q' \in S'$ (q is consumed by a broadcast) or $[S \setminus S' = \{p, q\}]$ and there exist two rules $\langle p, !!a, p' \rangle, \langle q, ??a, q' \rangle \in R$ such that $p', q' \in S'$ (p and q are consumed by a broadcast)].

Finally, we say that a set $S \subseteq Q$ satisfies an atom $\#q = 0$ if $q \notin S$ and it satisfies an atom $\#q \geq 1$ if $q \in S$; satisfiability for composite boolean formulae of CC is then defined in the natural way. We have then the following Lemma whose proof can be found in Appendix.

Lemma 2. *There is an execution of Algorithm 2 which answers YES on input \mathcal{P} and φ iff $\mathcal{P} \models \Diamond\varphi$.*

Proof. Let $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$ a process with $\text{MAHN}(\mathcal{P}) = \langle \mathcal{C}, \rightarrow, \mathcal{C}_0 \rangle$ and φ a constraint over \mathcal{P} in CC. First we assume that the Algorithm 2 answers

YES on input \mathcal{P} and φ . This means that there exists $S_0, \dots, S_m, T_0, T_1, \dots, T_n$ such that $1 \leq m, n \leq |Q|$ and $S_0 \subseteq Q_0$, and for all $i \in \{0, \dots, m-1\}$, $S_{i+1} \in \text{postAdd}(\mathcal{P}, S_i)$ and $T_0 = S_m$ and for all $i \in \{0, \dots, n-1\}$, $T_{i+1} \in \text{postDel}(\mathcal{P}, T_i)$. We will now prove that there exists two configurations $\gamma_0 \in \mathcal{C}_0$ and $\gamma \in \mathcal{C}$ such that $\gamma_0 \rightarrow^* \gamma$ and $L(\gamma) = T_n$. First, as reasoning the same way we did in the proof of Lemma 1, we can deduce that for any $k \in \mathbb{N} \setminus \{0\}$, there exists $\gamma_0 \in \mathcal{C}_0$ and a complete graph $\gamma_k = \langle V, E, L \rangle$ in \mathcal{C} such that $L(\gamma_k) = S_m$ and for every $q \in S_m$ the set $\{v \in V \mid L(v) = q\}$ has more than k elements. Now we are going to prove that for any $j \in \{0, \dots, n\}$, for all $k \in \mathbb{N} \setminus \{0\}$, there is a complete graph $\gamma_{j,k}$ such that:

1. $L(\gamma_{j,k}) = T_j$ and for each $q \in S_j$, the set $\{v \in V \mid L(v) = q\}$ has more than k elements (i.e. for each element q of S_j there are more than k nodes in $\gamma_{j,k}$ labelled with q),
2. there exists $\gamma_0 \in \mathcal{C}_0$ such that $\gamma_0 \rightarrow^* \gamma_{j,k}$.

To prove this statement we reason by induction on j . For $j = 0$, since the statement holds for S_m , it holds also for $T_0 = S_m$. We now assume that the property is true for all naturals smaller than j (with $j < n$) and we will show it is true for $j + 1$. We consider now the set $T_j \setminus T_{j+1}$ (assuming it is not empty, otherwise the property trivially holds). By property of the operator **postDel**, we have $T_{j+1} \subseteq T_j$. Now let $k \in \mathbb{N}$, the graph $\gamma_{j+1,k}$ is obtained from $\gamma_{j,k+1}$ as follows:

- if $T_{j+1} \setminus T_j = \{q\}$ and there exists a rule $\langle q, !!a, q' \rangle \in R$ such that $q' \in T_{j+1}$, then this rule is applied to all the nodes labelled by q ; first each node is isolated with the movement rule, then the broadcast rule is performed and then the complete graph is rebuilt. Note that the application of this rule consecutively will only increase the number of nodes labelled by q' which were already present in $\gamma_{j,N}$;
- if $T_{j+1} \setminus T_j = \{q\}$ and there exist two rules $\langle p, !!a, p' \rangle, \langle q, ??a, q' \rangle \in R$ such that $p, p', q' \in T_{j+1}$ (q is consumed by a broadcast), then all the nodes labelled by q are isolated together with a node labelled by p so that all these nodes are connected, then p broadcast a sending all the other nodes in q' and finally the complete graph is rebuilt; as a consequence there is no more nodes labelled by q , the number of nodes labelled by q' and p' have increased and the number of nodes labelled by p has decreased of one unit;
- if $T_{j+1} \setminus T_j = \{p, q\}$ and there exist two rules $\langle p, !!a, p' \rangle, \langle q, ??a, q' \rangle \in R$ such that $p', q' \in T_{j+1}$ (p and q are consumed by a broadcast), then as for the second case, we first eliminate all the nodes labelled by q by isolating them together with one node labelled by p , and then all the nodes labelled by p can be eliminated the same way it is done in the first case we considered.

By applying these rules it is then clear that $\gamma_{j,k+1} \rightarrow^* \gamma_{j+1,k}$ and also that $\gamma_{j+1,k}$ verifies the property 1 of the statement. Since by induction hypothesis, we have that there exists $\gamma_0 \in \mathcal{C}_0$ such that $\gamma_0 \rightarrow^* \gamma_{j,k+1}$, we also deduce that $\gamma_0 \rightarrow^* \gamma_{j+1,k}$, hence the property 2 of the statement also holds. Hence if

the Algorithm 2 returns YES on input \mathcal{P} and φ , we deduce that there exist a reachable configuration $\gamma \in \mathcal{C}$ such that $L(\gamma) = T_n$ and since T_n satisfies φ , we also have that $\gamma \models \varphi$, hence $\mathcal{P} \models \Diamond\varphi$.

We now assume that there exists two configurations $\gamma_0 \in \mathcal{C}_0$ and $\gamma \in \mathcal{C}$ such that $\gamma_0 \rightarrow^+ \gamma$ (the case $\gamma_0 = \gamma$ can be easily verified) and $\gamma \models \varphi$. Hence there exists $\gamma_1, \dots, \gamma_k \in \mathcal{C}$ such that $\gamma_0 \rightarrow^+ \gamma_1 \dots \rightarrow^+ \gamma_k$ with $\gamma_k = \gamma$ and for all $i \in \{1, \dots, k\}$, exactly one broadcast rule has been applied between γ_i and γ_{i+1} . From this execution we build a sequence of set of control states $(S'_i)_{0 \leq i \leq k}$ such that $S'_0 = L(\gamma_0)$ and for all $0 \leq i \leq k-1$, $S'_{i+1} = S'_i \cup L(\gamma_{i+1})$. By definition of the broadcast rule and of the operator **postAdd**, we deduce that $S'_{i+1} \in \text{postAdd}(\mathcal{P}, S'_i)$. From this sequence, we can furthermore extract a subsequence $(S_i)_{0 \leq i \leq m}$ such that for all $0 \leq i \leq m-1$, $S_{i+1} \in \text{postAdd}(\mathcal{P}, S_i)$ and $S_{i+1} \neq S_i$ and for all $0 \leq j \leq k$, there exists $0 \leq i \leq m$ such that $S'_j = S_i$. Since we have $S_i \subset S_{i+1}$ for all $0 \leq i \leq m-1$, we deduce that necessarily $m \leq |Q|$. Now we build another sequence of control states $(T'_i)_{0 \leq i \leq k}$ such that $T'_0 = S_m$ and for all $0 \leq i \leq k-1$, $T'_{i+1} = T'_i \setminus E_i$ where for all $0 \leq i \leq k-1$, $E_i = \{q \in L(\gamma_i) \mid \exists j > i \text{ s.t. } q \in L(\gamma_j)\}$. In other words, to build T'_{i+1} from T'_i we delete the control states q that are present in γ_i and will never be present in any γ_j for $j > i$. We recall that by construction for all $1 \leq i \leq k$, we have $L(\gamma_i) \subseteq T'_0$ and hence by construction of the sequence $(T'_i)_{0 \leq i \leq k}$ we have necessarily $L(\gamma) = T'_k$. By definition of the broadcast rule and of the operator **postDel**, we also deduce that $T'_{i+1} \in \text{postDel}(\mathcal{P}, T'_i)$. From this sequence, we can furthermore extract a subsequence $(T_i)_{0 \leq i \leq n}$ such that for all $0 \leq i \leq n-1$, $T_{i+1} \in \text{postDel}(\mathcal{P}, T_i)$ and $T_{i+1} \neq T_i$ and for all $0 \leq j \leq k$, there exists $0 \leq i \leq n$ such that $T'_j = T_i$. Since we have $T_{i+1} \subset T_i$ for all $0 \leq i \leq n-1$, we deduce that necessarily $n \leq |Q|$ and also we have $T(n) = L(\gamma)$. Since $\gamma \models \varphi$, we deduce that T_n satisfies φ and consequently we have proved that there is an execution of Algorithm 2 which answers YES on input \mathcal{P} and φ . \square

It is then clear that each check performed by the Algorithm 2 (i.e. $S_0 \subseteq Q_0$ and $S_{i+1} \in \text{postAdd}(\mathcal{P}, S_i)$ and $T_{i+1} \in \text{postDel}(\mathcal{P}, T_i)$ and T_n satisfies φ) can be performed in polynomial time in the size of the process \mathcal{P} and of the formula φ and since m and n are smaller than the number of control states in \mathcal{P} , we deduce the following theorem (the lower bound being given by Proposition 2).

Theorem 2. *PRP for constraints in CC is NP-complete.*

5 Complexity of the Cardinality Reachability Problem

In this section we study another problem, we call CRP, in which we ask the question whether we can reach a configuration with a given number of occurrences for each control state. Formally, given a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$, a *cardinality constraint* over \mathcal{P} is a function $\text{card} : Q \rightarrow \mathbb{N}$. We say that a configuration γ satisfies a cardinality constraint card (denoted by $\gamma \vdash \text{card}$) if for each $q \in Q$ the number of occurrences of q in γ is equal to $\text{card}(q)$. The Cardinality Reachability Problem (CRP) can then be stated as follows:

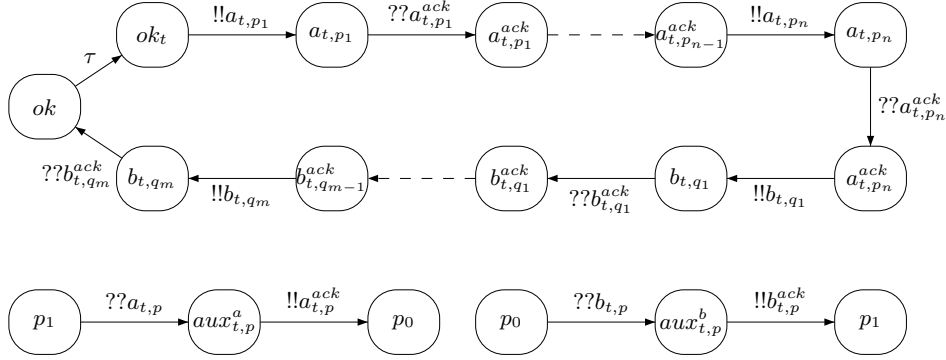


Fig. 1. Simulation of a transition t with $\bullet t = \{p_1, \dots, p_n\}$ and $t^\bullet = \{q_1, \dots, q_m\}$.

Input: A process \mathcal{P} with $MAHN(\mathcal{P}) = \langle \mathcal{C}, \rightarrow, \mathcal{C}_0 \rangle$ and a cardinality constraint $card$.

Output: Yes, if $\exists \gamma_0 \in \mathcal{C}_0$ and $\gamma_1 \in \mathcal{C}$ s.t. $\gamma_0 \rightarrow^* \gamma_1$ and $\gamma_1 \vdash card$.

Note that this problem seems easier than PRP because the cardinality constraint fixes the number of nodes of an initial configuration. In fact, if there is a reachable configuration which satisfies a cardinality constraint $card$, we know that this configuration and the initial configuration from which the computation starts have $\sum_{q \in Q} card(q)$ nodes. We will show that this is not the case as CRP is PSPACE-complete. First we prove the lower bound.

Proposition 3. *CRP is PSPACE-hard.*

Proof. We use a reduction from reachability in 1-safe Petri nets. A Petri net N is a tuple $N = \langle P, T, \mathbf{m}_0 \rangle$, where P is a finite set of places, T is a finite set of transitions t , such that $\bullet t$ and t^\bullet are multisets of places (pre- and post-conditions of t), and \mathbf{m}_0 is a multiset of places that indicates how many tokens are located in each place in the initial net marking. Given a marking \mathbf{m} , the firing of a transition t such that $\bullet t \subseteq \mathbf{m}$ leads to a new marking \mathbf{m}' obtained as $\mathbf{m}' = \mathbf{m} \setminus \bullet t \cup t^\bullet$. A Petri net P is 1-safe if in every reachable marking every place has at most one token. Reachability of a specific marking \mathbf{m}_1 from the initial marking \mathbf{m}_0 is decidable for Petri nets, and PSPACE-complete for 1-safe nets [2].

Given a 1-safe net $N = \langle P, T, \mathbf{m}_0 \rangle$ and a marking \mathbf{m}_1 , we encode the reachability problem as a CRP problem for the process \mathcal{P} and cardinality constraint $card$ defined next. For each place $p \in P$, we introduce control states p_1 and p_0 to denote the presence or absence of the token in p , respectively. Furthermore, we introduce a special control state ok . The control state is used to control the net simulation. Transitions of the controller are depicted in the upper part of Fig. 1. The first rule of the controller selects the current transition to simulate. The simulation of the transition t with $\bullet t = \{p_1, \dots, p_n\}$ and $t^\bullet = \{q_1, \dots, q_m\}$ is defined via two sequences of messages. The first one is used to remove the token from p_1, \dots, p_n , whereas the second one is used to put the token in q_1, \dots, q_m .

To guarantee that every involved place reacts to the protocol —i.e. messages are not lost— the controller waits for an acknowledgment from each of them. Transitions of places are depicted in the lower part of Fig. 1. It is not restrictive to assume that there is only one token in the initial marking \mathbf{m}_0 (otherwise we add an auxiliary initial place and a transition that generates \mathbf{m}_0 by consuming the initial token). Let p^0 be such a place. We define the initial states Q_0 of the process \mathcal{P} as $\{p_1^0, ok\} \cup \{p_0 \mid p \in P \setminus \{p^0\}\}$, in order to initially admit control states representing the controller, the presence of the initial token, and the absence of tokens in other places. The reduction does not work if there are several copies of controller nodes and/or place representations (i.e. p_1, p_0, \dots) interacting during a simulation (interferences between distinct nodes representing controllers/places may lead to incorrect results). However we can ensure that the reduction is accurate by checking the number of occurrences of states exposed in the final configuration: it is sufficient to check that only one controller and only one node per place in the net are present. Besides making this check, the cardinality constraint *card* should also verify that the represented net marking coincides with \mathbf{m}_1 . Namely, we define *card* as follows:

$$\begin{aligned} & \forall p \in \mathbf{m}_1, t \in T. \left(\begin{array}{l} card(p_1) = 1 \wedge card(p_0) = 0 \wedge \\ card(aux_{t,p}^a) = 0 \wedge card(aux_{t,p}^b) = 0 \end{array} \right) \wedge \\ & \forall q \notin \mathbf{m}_1, t \in T. \left(\begin{array}{l} card(q_1) = 0 \wedge card(q_0) = 1 \wedge \\ card(aux_{t,q}^a) = 0 \wedge card(aux_{t,q}^b) = 0 \end{array} \right) \wedge \\ & card(ok) = 1 \wedge \forall t \in T. card(ok_t) = 0 \wedge \\ & \forall t \in T, q \in P (card(a_{t,q}) = 0 \wedge card(b_{t,q}) = 0 \wedge card(a_{t,q}^{ack}) = 0 \wedge card(b_{t,q}^{ack}) = 0) \end{aligned}$$

Since the number of nodes stays constant during an execution, the post-condition specified by *card* is propagated back to the initial configuration. Therefore, if the protocol satisfies CRP for *card*, then in the initial configuration there must be one single controller node with state *ok*, and for each place p one single node with either state p_1 or state p_0 . Under this assumption, it is easy to check that a run of the protocol corresponds precisely to a firing sequence in the 1-safe net. Thus an execution run satisfies *card* if and only if the corresponding firing sequence reaches the marking \mathbf{m}_1 . \square

We now show that there exists an NPSpace algorithm to decide CRP. Let $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$. Since the size of a graph never changes during an execution, a cardinality constraint fixes the size of the initial configuration given by the sum K of constants in *card*. The algorithm guesses an execution $\gamma_0 \rightarrow \gamma_1 \rightarrow \dots \rightarrow \gamma_n$ traversing pairwise distinct configurations, s.t. γ_0 is a complete graph with K nodes in initial states, and then checks if *card* is satisfied in γ_n . Each configuration can be stored in polynomial space. Since the size of all configurations is K we need at most $K^{|Q|}$ (all possible combinations of states over K nodes). Thus we have a non-deterministic algorithm working in polynomial space. Since $\text{NPSpace} = \text{PSPACE}$, and in the light of the lower bound indicated by Proposition 3, we can conclude with the following theorem.

Theorem 3. *CRP is PSPACE-complete.*

6 Conclusion

We have studied the complexity of reachability problems for mobile ad hoc network protocols in which target states are represented by using constraints checking the presence, absence, or counting the number of occurrences of control states in a configuration. We have given algorithms for different classes of constraints. For constraints that simply checks the presence of control states we have shown that reachability is PTIME-complete, while when also constraints checking the absence are considered the problem turns out to be NP-complete. Finally, for constraints counting the number of occurrences reachability becomes PSPACE-complete. Our analysis significantly improves the decidability results given in [3] by reduction to problems which are known to be at least EXPSpace-hard.

References

1. Abdulla, P. A., Jonsson, B.: Verifying programs with unreliable channels. *Inf. Comput.* 127(2): 91–101 (1996)
2. Cheng, A., Esparza, J., Palsberg, J.: Complexity Results for 1-Safe Nets. *Theor. Comput. Sci.* 147(1&2): 117–136 (1995)
3. Delzanno, G., Sangnier, A., Zavattaro, G.: Parameterized verification of Ad Hoc Networks. *CONCUR’10*: 313–327
4. Delzanno, G., Sangnier, A., Zavattaro, G.: On the Power of Cliques in the Parameterized verification of Ad Hoc Networks. *FOSSACS’11*: 441–455
5. Ene, C., Muntean, T.: A broadcast based calculus for Communicating Systems. *IPDPS’01*: 149
6. Esparza, J., Finkel, A., Mayr, R.: On the verification of Broadcast Protocols. *LICS’99*: 352–359
7. Fehnker, A., van Hoesel, L., Mader, A.: Modelling and verification of the LMAC protocol for wireless sensor networks. *IFM’07*: 253–272
8. Godsken, J.C.: A calculus for Mobile Ad Hoc Networks. *COORDINATION’07*: 132–150
9. Ladner, R. E.: The circuit value problem is logspace complete for P. *SIGACT News*: 18–20 (1977)
10. Merro, M.: An observational theory for Mobile Ad Hoc Networks. *Inf. Comput.* 207(2): 194–208 (2009)
11. Merro, M., Ballardin, F., Sibilio, E. A Timed Calculus for Wireless Systems *FSEN’09*: 228–243
12. Lanese, I., Sangiorgi, D.: An operational semantics for a calculus for wireless systems. *TCS*, 411(19): 1928–1948 (2010)
13. Nanz, S., Hankin, C.: A Framework for security analysis of mobile wireless networks. *TCS*, 367(1–2):203–227 (2006)
14. Prasad, K.V.S.: A Calculus of Broadcasting Systems. *SCP*, 25(2–3): 285–327 (1995)
15. Saksena, M., Wibling, O., Jonsson, B.: Graph grammar modeling and verification of Ad Hoc Routing Protocols. *TACAS’08*: 18–32
16. Singh, A., Ramakrishnan, C. R., Smolka, S. A.: A process calculus for Mobile Ad Hoc Networks. *COORDINATION’08*: 296–314
17. Singh, A., Ramakrishnan, C. R., Smolka, S. A.: Query-Based model checking of Ad Hoc Network Protocols. *CONCUR ’09*: 603–619

18. Rackoff C.: The Covering and Boundedness Problems for Vector Addition Systems. TCS, 6:223-231 (1978)
19. Lipton R.J.: The Reachability Problem Requires Exponential Space. Department of Computer Science. Research Report. Yale University. (1976)